DOI:
10.1016/j.cirp.2012.03.051

Publication date:
2012

Document Version
Peer reviewed version

Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Modeling of machine tools using smart interlocking software blocks

Aydin Nassehi, Stephen T Newman (2)

Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK

Abstract
Machine tools are traditionally designed to maximize performance, precision and repeatability of manufacturing processes. New criteria for design including maximizing energy efficiency and reconfigurability are now emerging. In this paper, a novel methodology is proposed for representing machine tool elements as smart interlocking software blocks that are dynamically structured based on predefined ontology and then combined to form a holistic model of a machine tool. This model can be used to assess, simulate and optimize the machine tool against a range of criteria. A prototype implementation of the methodology is demonstrated using two test cases for kinematics and power usage.

Keywords: Modeling, Machine Tool, Emergent Synthesis.

1. Introduction

Most of the production techniques utilized in manufacturing today rely on machine tools and their use at some point in their lifecycle. It is therefore imperative to perform various types of analysis, simulation and optimization throughout the lifecycle of machine tools to ensure there most economical use is achieved.

Traditional methods for analysis of machine tools focus on maximizing performance, precision and repeatability of manufacturing processes. However, with emerging trends towards sustainability and the invention of numerous new technologies such as hybrid manufacturing techniques, the existing analysis frameworks are no longer sufficiently capable of meeting the requirements of the entire range of analysis required throughout the make, use, maintain and recycle phases of machine tools.

In this paper, the simplification of various types of analysis carried out on machine tools is first explored. This is used to ascertain the divisibility of these problems into smaller similarly structured problems in conjunction with models that are used to create the abstract domain for specific types of analyses. A framework based on smart interlocking software blocks – a context aware, ontology based mutually referential, object-oriented model – is then proposed for synthesis of holistic machine tool models. A prototype implementation of the modeling approach is then used with two test cases to demonstrate the viability of the methodology for use in manufacturing problems.

2. Machine tool models and their applications

In order to perform any type of analysis, optimization or simulation, an abstract model is required [1]. With regards to manufacturing resources, in general, and machine tools, in particular, various types of models with a wide range of fidelity are used for different applications.

Mathematical models reduce the machine tool into a mathematical equation, or a system of equations, that represent a single aspect of the machine tool. Examples of these models include Denavit-Hartenberg (D-H) matrices used for kinematic analysis [2] and formulations used for modeling cutting process parameters [3,4]. The common attribute of these models is that the semantics of the various variables are not stored within the model, that is, validation of the data that is used within the mathematical relationship is carried out separately from the model. It is thus assumed that the context (i.e. ontology and aims) of the model is specified *a priori*. It is therefore necessary, to have a complicated abstraction process to determine the parameters required in the mathematical model from the physical entities in the real world.

Models with an associated meta-model are models that are generally constructed to simplify the parameter abstraction process for storing the various parameter values required by mathematical models in a meaningful manner.

These models are regulated through the use of meta-models that allow the relationship between various items of data to be stored as well as the data itself. Manufacturing resource data models are examples of this type of models [5,6]. These models are used either on their own for cataloguing purposes or in conjunction with various mathematical models to enable computer aided manufacturing resource analysis [7]. Graphical machine tool models such as those used for cutting simulation (e.g. in Vericut) belong to these types of models where information about shapes is linked together with kinematic equations to enable the simulation of movement.

The most intricate models are **ontology-based models**. In these models, in addition to the information and meta-data for structuring the information, the semantics of the knowledge stored within the model is also stored. Such models have been extensively used in design since the 1980s [8] but within the realm of machining, only recently, machine tool models based on this methodology are being introduced [9].

To better illustrate these three types of models, an example of a simple mechanical system is provided in figure 1.

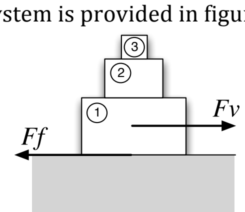


Figure 1. A simple mechanical system for showing the difference between three types of modeling.

In this example, a number of solids are joined on top of each other and the aim is to calculate the maximum friction force (F_f) against movement when a horizontal force (F_v) is applied to the system based on the coefficient of friction (μ), the masses and Earth's gravity. For such a simple system, the models that are useful for calculation of the maximum friction force have been summarized in Table 1.

The ontology-based models offer the advantage of being able to store information required for multiple types of analysis in the same model. This results in less redundancy in storage of information, better cohesion between various calculations and ultimately enabling new types of cross-technology and cross-purpose analysis. In this paper a computational framework for implementing an ontology-based model of machine tools for multi-criteria analysis is proposed that allows the advantages of ontology based modeling, including dynamic structural updates and thus repurposing capability of the models, to become available for machine tool analysis.

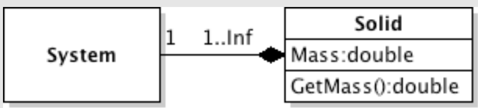
3. Using smart interlocking software blocks to create models useful for multi-criteria analysis

One of the important advantages of ontology based modeling is that regulated synthesis - combination of local knowledge to form global knowledge according to pre-defined rules - becomes achievable. Where current methods of modeling machine tools are based, solely, on analysis, the approach of emergent synthesis will allow advanced attributes to emerge globally, based on locally defined interactions [10]. With the existence of the ontology, it will be possible to regulate the synthesis in a dynamic manner to ensure that the emergent attributes are representative of the real system, in this case a machine tool, even when full information about the domain is not available.

3.1 Ontological foundation

Many tools have been proposed for capturing manufacturing ontologies with varying success for particular purposes. In order to allow systemic evolution of production to be modeled using the proposed computational framework, a methodology that allows dynamic updates in the ontology is required. The web ontology language (OWL), in particular, provides strong tools for underpinning semantics of a wide range of domains and due to its popularity [11], use of the ubiquitous XML and ease of interface with modern programming languages, it has been chosen to model the ontological foundations required for synthetic modeling of machine tools using smart interlocking blocks.

Table 1 Models for calculating maximum friction force.

Modeling methodology	**System Representation**
Mathematical Model	$F_f = \mu g(m_1 + m_2 + m_3)$
Model with a Meta-Model	 $F_f = \mu g(S.Solid[1].mass + S.Solid[2].mass + S.Solid[3].mass)$
Ontology-based Model	**- Solid** *has* **mass** **- Join**ing **Solid**s *results in a* **Solid** *with* **mass** *equal to the sum of the* **mass**es *of the* **Solid**s that are joined together. - **FrictionForce** *for a* **Solid** *is* **mass** *multiplied by μg.* **- System** *is a number of solids* **join**ed.

3.2 Emergent synthesis in software blocks

With the existence of the manufacturing ontology, the specification of the artifactual system becomes complete and as the purpose of modeling is to capture the information in the environment, completeness of the information is an assumption for a deterministic model. Thus, the emergent synthesis problem can be classified as a Class I problem according to the classification provided in [10]. Figure 2 shows the approach chosen for emergent synthesis of the machine tool model.

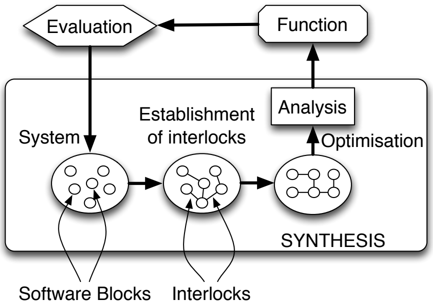


Figure 2. Machine tool modeling as a Class I emergent synthesis problem - adapted from Ueda et al. [10]

3.3 Software block structure

In order to achieve the emergent synthesis, a modular structure of interconnected entities is chosen for the system. Similar system structures have been used in the modeling of processes with function blocks [12] and in the use of multi agent systems in production [13,14] but these implementations have static structures for each module that are not dynamically updated.

Object-orientation provides a suitable implementation platform for building such a system. The main advantage of this architecture is that different entities in the system are modeled using very similar blocks and as such different components can be addressed and manipulated using the same software routines. This provides a simpler architecture for manufacturing software that performs the analysis on the system. The structure of the fundamental software block used within the proposed system to model a component of the machine tool is shown in Figure 3.

Each software block stores a copy of the ontology that allows local access to the underlying semantics of the model. Based on this local copy of the ontology, it is possible to define an identity for each software block.

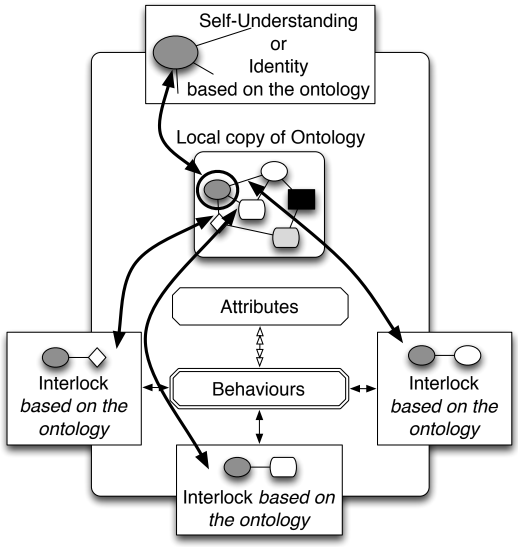


Figure 3. Anatomy of a Smart Interlocking Software Block (SIB)

This identity allows the software block to maintain an understanding of itself and the interlocks that connect it to other software blocks. The software block also stores the attributes associated with the respective machine tool component that it represents. A representation of the actions that the component performs also resides within the software block. For example a software block that models a motor would contain attributes such as the motor power, mass and dimensions as well as a model of the action of turning the motor on to exert torque.

3.4 Interlocking capabilities based on the defined ontology

The second element of the proposed system is the interlocks. These logical entities enable and regulate the exchange of information between various software blocks. Having interlocks separate from the software blocks, allows the creation of dynamic links when necessary and also enable changes and evolution in the system according to changes in the ontology of the system.

The interlocks will allow the software blocks to be linked in serial and parallel configurations or to form closed loops structures, if allowable according to the ontology. Instantiation of the model takes place in a software environment where the user selects the appropriate software blocks for modeling the intended machine and drops them in a work area. Interlocks are suggested automatically based on the ontology and approved by the user to create the complete model of a machine.

3.5 Defining context sensitive properties to make smart software blocks

With the existence of dynamic interlocks in the machine tool model, it is possible to incorporate context sensitivity in the functionality of the interlocking software blocks. This means that an existing model can be repurposed for a new type of analysis with minimal effort. Consider the example scenario where a machine tool model created using the proposed approach for kinematic calculations is to be adapted for calculating forces in the machine structures. The first step for this repurposing is to add the necessary constructs (i.e. masses, friction factors, etc.) to the ontology. The software blocks will then update their structure and the attributes are added to each block. The interlocks will also be updated with the new capabilities as described in the ontology. The second and final step to finish repurposing is to assign the values for each attribute to the software blocks.

3.6 Consolidating the computational framework with a mathematical platform to solve synthesized systems of equations

In order to carry out analysis on the created machine tool model, first the mathematical relationships stored in software blocks and interlocks are synthesized to form a coherent set of equations representing a holistic aspect of the machine tool (e.g. the kinematic equations or equations for calculating energy consumption in machining). Solving these equations will require a mathematical analysis engine. In the proposed system, a constraint modeler, capable of optimizing and solving complex systems of equations has been utilized. For details about the constraint modeler and its use refer to [15]. The overview of the proposed modeling approach is shown in figure 4.

When a query is received by the model (i.e. asking the model how much power would a manufacturing operation require or what the state of a system should be for the tool to be at a specific coordinate with a defined orientation) it is passed on to a smart interlocking software block. The software blocks and the interlocks pass the query along. Each block and interlock generates the mathematical equation that governs the particular element or interface that it represents.

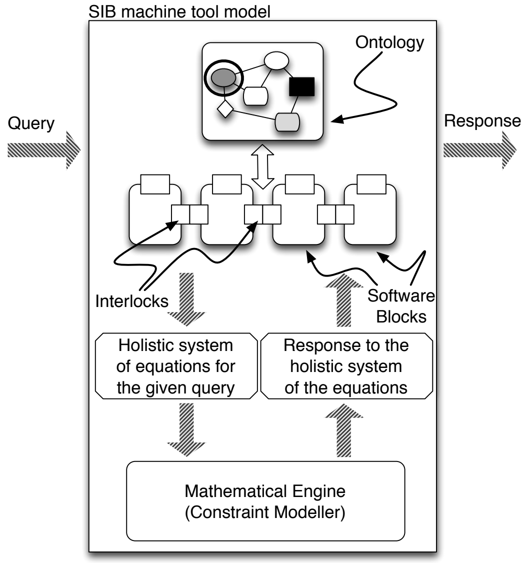


Figure 4. Overview of the modeling approach in response to queries

This system of equations is then passed on to the mathematical engine that solves the system and passes the results back to the software blocks. The blocks interpret the results and form the response to the query and the model returns the appropriate response.

4. Test cases

A prototype implementation of the modeling approach has been realized using the Java programming language with an OWL based ontology together with the constraint modeler.

This implementation has been used with two models to illustrate the advantages of the approach. First the simple 2D robot shown in Figure 5 is used to illustrate the entire modeling approach and then a more complex model of a machine tool is used to show the applicability of the approach in practical scenarios.

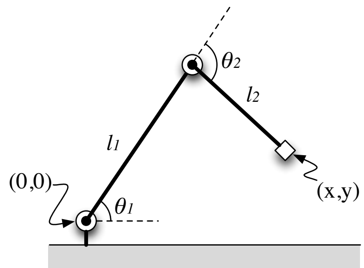


Figure 5. Simple 2D robot with 2 degrees of freedom

The smart interlocking software blocks for the model of the 2D robot are shown in Figure 6. Six software blocks are created to represent the two links, the two motors, the ground connection and the effector. Five interlocks represent the links between these components. A query is formed to assess what the state of the components of the machine need to be for the effector to be at the position (*1,0.5*) where $l_1=1$ *and* $l_2=0.8$. The resulting equations are shown in table 2. The constraint modeler successfully solves these equations to find the answer (*θ_1=-17.4°, θ_2=-104.11°).*

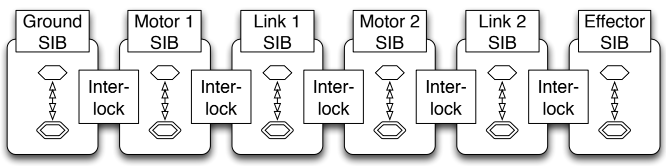


Figure 6. SIB model of the 2D robot

These equations can be easily pre-optimized by renaming variables that are constrained to be equal to each other to the same variable. This results in reduction of equations to the same number as the degrees of freedom existing in the model and thus this approach is no more complex than other mathematical approaches such as the D-H matrices for any given kinematic system.

To assess the practical capability of the modeling approach the simplified the model of a vertical machining center as shown in figure 7 is used to estimate the average power required to machine a series of slots in an aluminum block. The experimental results for the machining were reported in [16]. The power requirement by the model is compared to the average actual power for different depths of cut in table 3; this shows that the difference between the results is less than 5%.

Table 2 Equations generated by the 2D machine tool model

Code	Generated by
dec real Ground_x, Ground_y, Ground_a;	Ground SIB
dec real Motor1_x,Motor1_y,Motor1_a,Motor1_t;	Motor 1 SIB
dec real Link1_x,Link1_y,Link1_a,Link1_l;	Link 1 SIB
dec real Motor2_x,Motor2_y,Motor2_a,Motor2_t;	Motor 2 SIB
dec real Link2_x,Link2_y,Link2_a,Link2_l;	Link 2 SIB
dec real Effector_x,Effector_y,Effector_a;	Effector SIB
dec real xpos,ypos;	Query
Ground_x=0; Ground_y=0; Ground_a=0;	Ground SIB
Link1_l=1;	Link 1 SIB
Link2_l=0.8;	Link 2 SIB
xpos=1; ypos=0.5;	Query
function solve	
{ var Motor1_x,Motor1_y,Motor1_a,Motor1_t;	Motor 1 SIB
var Link1_x,Link1_y,Link1_a;	Link 1 SIB
var Motor2_x,Motor2_y,Motor2_a,Motor2_t;	Motor 2 SIB
var Link2_x,Link2_y,Link2_a;	Link 2 SIB
var Effector_x,Effector_y,Effector_a;	Effector SIB
rule (Effector_x - xpos); rule (Effector_y - ypos);	Query
*rule (Effector_x - Link2_x-Link2_l*cos(Link2_a));*	Effector-Link2
*rule (Effector_y - Link2_y-Link2_l*sin(Link2_a));*	Interlock
rule (Effector_a - Link2_a);	
rule (Link2_x-Motor2_x);	
rule (Link2_y-Motor2_y);	Link2-Motor2
rule (Link2_a-Motor2_a-Motor2_t);	Interlock
*rule (Motor2_x - Link1_x-Link1_l*cos(Link1_a));*	
*rule (Motor2_y - Link1_y-Link1_l*sin(Link1_a));*	Motor2-Link1
rule (Motor2_a - Link1_a);	Interlock
rule (Link1_x - Motor1_x);	
rule (Link1_y - Motor1_y);	Link1-Motor1
rule (Link1_a - Motor1_a - Motor1_t);	Interlock
rule (Motor1_x - Ground_x);	
rule (Motor1_y - Ground_y);	Motor1-Ground
rule (Motor1_a - Ground_a);}	Interlock

5. Conclusions

The modeling approach presented in this paper allows smart interlocking software blocks representing individual components of machine tools to be synthesized to form holistic models of machine tools.

These blocks are then capable of producing the mathematical system of equations representing various aspects of the machine.

As the structure of the software blocks is dynamically updated according to the specified ontology, the model can be easily repurposed to perform functions beyond its original purpose allowing multi-criteria analysis to be performed with minimal effort.

Table 3 Estimated machining power compared with actual power used.

Cutting Depth	Estimated power kW	Actual power (±0.03) kW	Difference Percentage
1mm	*3.12*	*3.28*	*4.8%*
2mm	*3.23*	*3.37*	*4.1%*
3mm	*3.29*	*3.42*	*3.8%*
4mm	*3.37*	*3.48*	*3.2%*

References

[1] Freudenthal, H., 1961, The concept and the role of the model in mathematics and natural and social sciences, Kluwer Academic Pub.

[2] Harib, K., Ullah, A.S., Hammami, A., 2007, A hexapod-based machine tool with hybrid structure: Kinematic analysis and trajectory planning, Int. J. of Machine Tools and Manufacture, 47/9:1426-1432.

[3] Van Luttervelt, C., Childs, T., Jawahir, I., Klocke, F., Venuvinod, P., Altintas, Y., Armarego, E., Dornfeld, D., Grabec, I., Leopold, J., Lindstrom, B., Lucca, D., Obikawa, T., Shirakashi, Sato, H., 1998, Present situation and future trends in modeling of machining operations: progress report of the CIRP working group 'modelling of machining operations', CIRP Annals – Manufacturing Technology, 47/2:587-626.

[4] Paris, H., Brissaud, D., Gouskov, A., 2007, A more realistic cutting force model at uncut chip thickness close to zero, CIRP Annals – Manufacturing Technology, 56/1:415-418.

[5] Vichare, P., Nassehi, A., Kumar, S., Newman, S.T., 2009, A unified manufacturing resource model for representing CNC machining systems, Robotics and Computer Integrated Manufacturing, 25/6:999-1007.

[6] Pullan, T., Bhasi, M., Madhu, G., 2012, Object-oriented modeling of manufacturing information system for collaborative design, Int. J. of Production Research, in press.

[7] Altintas, Y., Brecher, C., Weck, M., Witt, S., 2005, Virtual machine tool, CIRP Annals – Manufacturing Technology, 54/2:115-138.

[8] Liu, Y., Lim, S., 2010, Using ontology for design information and knowledge management: A critical review, Global Product Development: CIRP Design Conference, 427-433.

[9] Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G., Chen, D., 2009, The machine tool model – a core part of the digital factory, CIRP Annals – Manufacturing Technology, 58/1:425-428.

[10] Ueda, K., Markus, A., Monostori, L., Kals, H., Arai, T., 2001, Emergent synthesis methodologies for manufacturing, CIRP Annals – Manufacturing Technology, 50/2:535-551.

[11] Delamer, I., Martinez Lastra, J., 2011, Ontologies of production automation, IFIP Lecture Notes in Computer Science (LNCS), 4891:276-289.

[12] Wang, L., Song, Y., Gao, Q., 2009, Designing function blocks for distributed process planning and adaptive control, Engineering Applications of Artificial Intelligence, 22/7:1127-1138.

[13] Monostori, L., Vancza, J., Kumara, S., 2006, Agent-based systems for manufacturing, Annals of the CIRP, 55/2:697-720.

[14] Teti, R., Kumara, S., 1997, Intelligent computing methods for manufacturing systems, CIRP Annals – Manufacturing Technology, 46/2:629-652.

[15] Mullineux, G., 2001, Constraint resolution using optimisation techniques, Computers & Graphics, 25/3:483-492.

[16] Newman,S.T., Nassehi, A., Imani-Asrai, R., Energy efficient process planning for CNC machining, 1st Int. Colloquium of the Cluster of Excellence (eniPROD): Energy efficient product and process innovation in production engineering, June 2010.

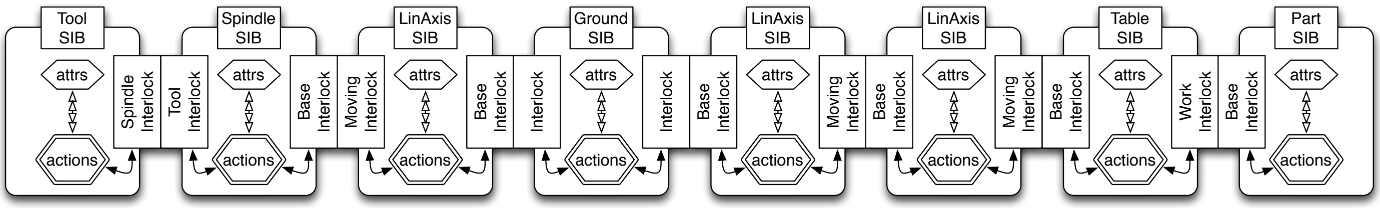


Figure 7. Simplified SIB model of a 3-axis vertical machining center